Countermeasures for Mitigating ICN Routing Related DDoS Attacks

Eslam G. AbdAllah(✉), Mohammad Zulkernine, and Hossam S. Hassanein

Queen's University, 99 University Ave, Kingston, ON K7L 3N6, Canada
{eslam,mzulker,hossam}@cs.queensu.ca

Abstract. Information Centric Networking (ICN) is a new communication paradigm for the future Internet that focuses on contents rather than infrastructures or end-points. Distributed Denial of Service (DDoS) attacks that may occur in many scenarios in an ICN, can overwhelm ICN routing and caching resources. In this paper, we focus on routing related DDoS attacks from both publisher and subscriber points of view and how they impact ICNs. We then propose a generic solution independent of a specific ICN architecture. This solution is based on a number of countermeasures: request satisfaction ratio, request rate limit, rating for contents and publishers, and test message. We present the implementation results, which show that the solution mitigates the routing related DDoS attacks and efficiently enhances the ICN performance in the existence of these attacks.

Keywords: Information centric networking · Distributed denial of service · ICN routing.

1 Introduction

The Internet was originally developed in the 1970's as the Internet of hosts. Nowadays, the Internet appears as Internet of things, Internet of services, Internet of people and Internet of media. According to Cisco Visual Networking Index 2013, global IP traffic per month will reach approximately 126 Exabytes by the year 2017 [1]. Information Centric Networking (ICN) is one of the proposed alternatives for these new Internets and requirements. ICN mainly depends on location-independent naming, name-based routing, built-in security and in-network caching [2]. The most popular ICN architectures are Named Data Networking (NDN), Data Oriented Network Architecture (DONA), Network of Information (NetInf), and Publish Subscribe Internet Technology (PURSUIT) [3]. All ICN architectures share several common components: information object, naming, routing, caching, security, and application programming interface.

This paper investigates the routing related Distributed Denial of Service (DDoS) attacks in an ICN in general regardless of a specific ICN architecture. We address seven different scenarios in which the attacker can be a malicious subscriber or a malicious publisher or both.

J. Tian et al. (Eds.): SecureComm 2014, Part II, LNICST 153, pp. 84–92, 2015.
DOI: 10.1007/978-3-319-23802-9_9

The proposed solution consists of five countermeasures. First, the request satisfaction ratio (RSR) measures the ratio between satisfied and outgoing requests per ICN router interface. RSR depends on a one-to-one relation between the request and the response in an ICN architecture. Second, the request rate limit applies rate limitations for ICN requesters that exceed the request rate thresholds. The ranking for ICN contents (third) and publishers (fourth) mitigates the effects of malicious contents and publications. Fifth, test message is utilized to check the validity of announced routes. To evaluate our solution, we implement the solution on ndnSIM [4]. The ndnSIM is a simulator for Named Data Networking (NDN) architecture and it is an NS-3 based module. The suggested solution is specifically tailored for various unique aspects of the ICN. The in-network caching is one of the major ICN components and not available in non-ICN environments. There are no host addresses; therefore the solution does not depend on any IP-based addressing as in non-ICN environments. The request satisfaction ratio depends on the ICN property that each request has only one response and there is no response without a request. However, in non-ICN environments, a request can receive many data packets. The rating for ICN contents ranks the contents regardless of its source. The other two countermeasures (request rate limit and rating for publisher) depend on the RSR.

2 Attack Scenarios

In this section, we present a comprehensive list of routing related DDoS attack scenarios in the ICN. We used a generic model of an ICN architecture as a reference model. This model consists of ICN routers, distributed storage location, and ICN users. ICN routers contain routing and caching capabilities. The distributed storage locations are used to store the ratings for ICN contents and publishers. ICN users are classified into publishers and subscribers. ICN subscribers can send a subscription message or vote against an invalid content.

An attacker can overwhelm the ICN resources such as bandwidth, routing tables, processing, and storage in the following scenarios:

1. Attacker sends malicious requests for available contents. (subscriber)
   (a) For the same content.
   (b) For different contents.
2. Attacker sends malicious requests for unavailable contents. (subscriber)
3. Attacker sends malicious requests for available and unavailable contents. (subscriber)
4. Attacker announces invalid routes. (publisher)
5. Attacker announces invalid contents. (publisher)
6. Attacker votes against valid contents. (subscriber)
7. Attacker announces invalid contents and another attacker requests for invalid contents and does not vote against these contents. (publisher and subscriber)

An attacker can be a malicious subscriber or a malicious publisher or both as indicated after each scenario listed above. The impacts of these attacks may be

amplified if the attackers act in a distributed manner. Scenario 1.a does not need any special countermeasure in an ICN, as there is in-network caching that can respond from an access router connected to subscribers. Scenario 1.b is similar to scenario 7. The main difference is the practical difficulty of scenario 1.b, as an attacker needs to send many different requests for available contents. In scenario 7, a malicious publisher announces invalid contents and a malicious subscriber requests for them dynamically. Scenario 4 causes the same impacts of the request timeout as scenario 2. As a part of our solution depends on user voting against malicious publications, thus scenario 6 is included.

Some existing works address malicious subscriptions [5–7]. They work on a specific ICN architecture and each one of them addresses a specific type of DDoS attack. Gasti et al. [8] present a high level classification of DDoS attacks and their solutions in NDN. Some other papers also classify DDoS attacks and their detection/prevention mechanisms in general [9–11]. The famous countermeasures for DDoS in the Internet architecture are IP trace back, packet filtering, and rate limiting. These techniques cannot be used in the ICN as they depend on IP addresses for the end-points.

3 Countermeasures

The proposed solution consists of five countermeasures for ICN routing. When the subscriber sends a request, an ICN router checks its cache, and if the requested content is not in the cache it forwards the request to the ICN. The ICN tries to get the best available content with the best trusted publisher based on their ranking stored in the distributed storage. The ICN router forwards the response to the subscriber and updates the request satisfaction ratio for this user. The request rate limit, rating for contents, and rating for publishers countermeasures are dependent on the RSR. The subscriber also can vote against an invalid content. The publisher sends an announcement for his/her content route. The RSR, request rate limit, rating for contents, and rating for publishers handle the attack scenarios 1, 2, 3, and 7. The test message addresses the attack scenario 4. The RSR, rating for contents, and rating for publishers handle the attack scenarios 5 and 6. This solution is implemented with a pushback mechanism that allows ICN routers to cooperate for achieving a better performance [5,6]. The countermeasures are as follows:

Request Satisfaction Ratio (RSR): RSR measures the number of the satisfied requests with respect to the number of outgoing requests. The request satisfaction ratio for interface i (RSR_i) is calculated by the following equation:

$$RSR_i = \frac{number\ of\ satisfied\ requests}{number\ of\ outgoing\ requests} \tag{1}$$

ICN architectures can manage a distributed storage depending on whether the architecture contains a name resolution entity or not. In the architectures with a name resolution entity (e.g., DONA, NetInf), the vote can be directed to

the connected name resolution entity. The name resolution entity then updates the other name resolution entities as a normal publication process in these architectures. Each entry in the name resolution entity contains the ratings for the contents and publishers in addition to the normal record. In the architectures without a name resolution entity, there are also two other situations. When the architecture has global locations for publication and subscription like the PUSURIT architecture, the votes can be directed to the connected rendezvous network and the interconnected networks update the ratings for contents and publishers. When there is no centralization of any sort like the NDN architecture, we need a storage capability such as distributed databases, distributed hash table or cloud-based solutions to store the ratings for contents and publishers.

The solution incorporates three messages to the ICN API primitives "publish" and "subscribe" as follows: (1) vote message: subscriber votes against a certain content. It uses the content name as the main parameter, (2) alert message: ICN router sends an alert to a subscriber when a content or publisher rate is more than a certain threshold value, and (3) test message: ICN router sends a test message through a route that does not return any response to check whether this route is malicious or not.

Request Rate Limit: This countermeasure limits the incoming requests based on the traffic rate and the request satisfaction ratio. For a given time interval, if the number of requests from interface i exceeds a certain threshold limit, then the ICN limits the incoming requests from interface i by:

$$Request\ rate\ limit = \frac{RSR_i * L_{max}}{n} \tag{2}$$

where L_{max} is the maximum number of routing table entries and n is the total number of the interfaces.

Rating for Contents: This countermeasure ranks ICN contents, consequently an ICN can select the best trusted available content. The voting weight against content $c\,(W_{content})$ is calculated by the following equation:

$$W_{content} = \sum_{i=1}^{n} \frac{CV_{Ui}}{number\ of\ U_i\ votes} * RSR_i \tag{3}$$

where U_i is the user who connected to interface i, CV_{U_i} is the U_i votes against content c, and n is the number of votes against this content. The voting ratio against content $c\,(R_{content})$ is calculated based on the following equation:

$$R_{content} = \frac{number\ of\ votes\ against\ (c)}{number\ of\ downloads\ for\ (c)} \tag{4}$$

From Eqs. (3) and (4), we drive the following equation:

$$Rating\ for\ content\ (c) = W_{content} * R_{content} \tag{5}$$

Rating for Publishers: This countermeasure ranks ICN publishers. As a result, an ICN can select the best trusted publisher. The voting weight against publisher $p\,(W_{publisher})$ is calculated by the following equation:

$$W_{publisher} = \sum_{i=1}^{n} \frac{PV_{Ui}}{number\ of\ U_i\ votes} * RSR_i \tag{6}$$

where U_i is the user who connected to interface i, PV_{U_i} is the U_i votes against publisher p, and n is the number of votes against this publisher. The voting ratio against publisher $p\,(R_{publisher})$ is calculated by the following equation:

$$R_{publisher} = \frac{number\ of\ publications\ that\ received\ voting}{number\ of\ publications\ from\ (p)} \tag{7}$$

From Eqs. (6) and (7), we get the following equation:

$$Rating\ for\ publisher\ (p) = W_{publisher} * R_{publisher} \tag{8}$$

Figure 1 presents the flowchart of the proposed solution. The flowchart uses two functions (check request availability and user voting) that are shown in Algorithm 1 and 2, respectively. This solution is implemented in ICN routers. Once the subscriber sends a request, the router checks its availability. If this request passes the availability check, the router forwards it and waits for the response (assuming the requested content is not in the cache). Then the ICN finds the best available content based on our ranking to ICN contents and publishers. An ICN router checks the response in three cases. First, when the content is found, the router updates the RSR for this interface and checks the rating for ICN contents and publishers. If the rating is more than a certain threshold value, it sends an alert message to the subscriber, who decides whether to accept this content or not. If the subscriber receives a content, he/she can vote against it. Second, when there is no content found with the requested name, the router just updates the RSR for this interface. Third, when the request is timed out, the router first checks the behavior of this interface. If the RSR of this interface is under a certain value, then the router directly updates the RSR for this interface. If the RSR is above the threshold value, the router sends a test message to check the announced route. If the router gets a response, then it retransmits the request again. Otherwise, it marks this route as a malicious one and also updates the RSR. All the threshold values can be dynamically set by ICN administrators. In Algorithm 2, the router detects whether this request is legitimate or not by checking the RSR and request rate for an interface.

Algorithm 1 describes the user voting against a false content. Then ICN routers send the voting message with the calculated weight to the storage location.

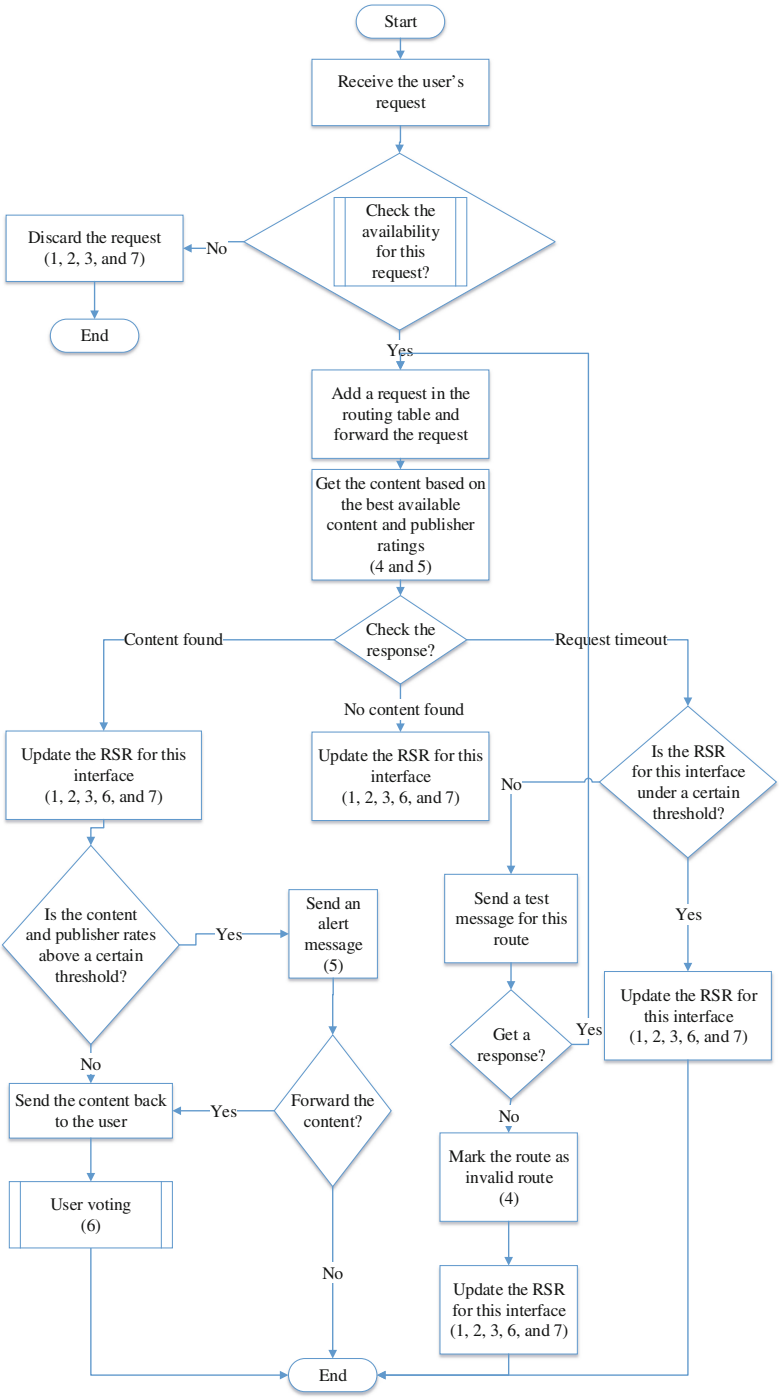

Fig. 1. Solution steps and countermeasures for ICN routing related DDoS attacks scenarios (the numbers inside the boxes indicate the affected scenarios)

Algorithm 1. User voting

Input: Received content upon users request
1: **if** *content* is *invalid* **then**
2: send vote message
3: update the rating for the content and publisher by the ICN
4: **end if**

Algorithm 2. Check request availability

Input: Incoming users request via interface i
1: **if** RSR_i is *valid* **then**
2: **if** request rate $<$ threshold limit **then**
3: **return** Yes
4: **else**
5: **return** No
6: **end if**
7: **else**
8: **return** No
9: **end if**

4 Implementation and Results

In this section, we study the impacts of ICN routing related DDoS attacks with and without the proposed solution. We evaluate the solution using the ndnSIM as a proof of concept. We build our experiments using the AT&T network which is Internet-like real network. Our implementation parameters are as follows: no.

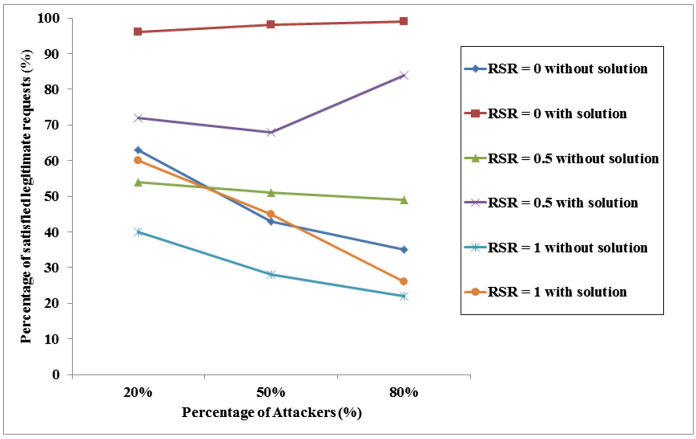

Fig. 2. Percentage of satisfied legitimate requests in the existence of attack scenarios 1, 2, 3, and 7 with and without the proposed solution

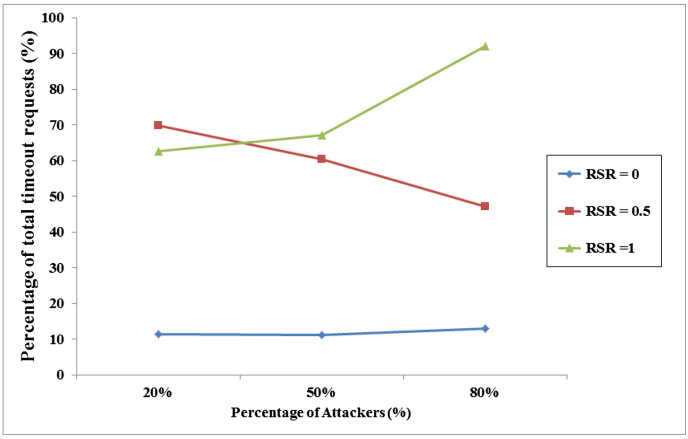

Fig. 3. Ratio of timeout requests with and without the proposed solution in all ICN nodes in the existence of attack scenarios 1, 2, 3, and 7

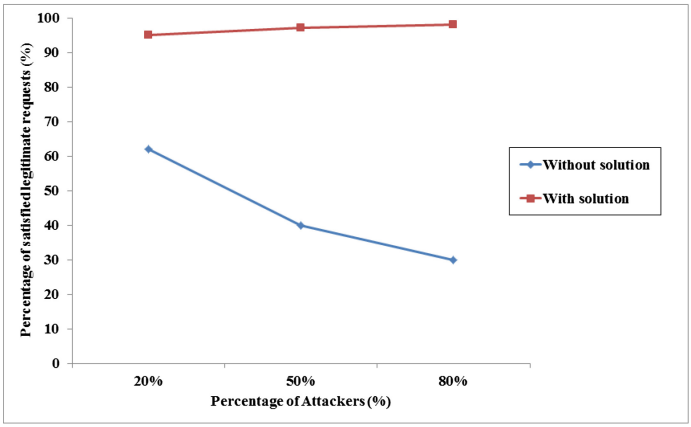

Fig. 4. Percentage of satisfied legitimate requests in the existence of attack scenario 4 with and without the proposed solution

of requests/s for the legitimate user = 100, no. of requests/s for the attacker = 1000, no. of subscribers = 20, no. of publishers = 3, no. of publishers in evaluating the test message = 20, packet size = 1 Kbytes, Pending Interest Table (PIT) size = 1000, cache size = 1000. We used the default values for any other parameters. There are three cases with different RSRs: RSR = 0 for unavailable content requests; RSR = 0.5 for 50 % unavailable content requests and 50 % different available content requests; RSR = 1 for different available content requests. We perform these experiments when the percentage of the attackers to the legitimate users are 20 %, 50 %, and 80 %. As depicted in Fig. 2, the solution mitigates the routing related DDoS attacks and enhances the ICN performance in the three

cases. When RSR = 0.5, as the number of attackers increases more than 50 %, the solution achieves better performance. This happens because the solution limits more requests from the attackers. As shown in Fig. 3, the solution also decreases the percentage of timeout requests in the three cases. These experiments cover scenarios 1, 2, 3, and 7. Figure 4 shows that the results of the impact of scenario 4 are close to the impact of scenario 2. The minor difference between the two impacts comes from the extra overhead due to test messages. For attack scenarios 5 and 6, the solution ranks ICN contents and publishers, which makes these attack scenarios difficult and also lessens their impacts on ICN.

5 Conclusion

ICN is one of the proposed architectures for the future Internet. DDoS attacks have significant impacts on ICN resources. In this paper, we present different scenarios of routing related DDoS attacks that may happen in an ICN. We also present our proposed generic solution, which consists of five countermeasures. The solution enhances the ICN performance in all attack cases.

References

1. Cisco visual networking index: forecast and methodology, 2012–2017 (2013)
2. Pan, J., Paul, S., Jain, R.: A survey of the research on future internet architectures. IEEE Commun. Mag. **49**(7), 26–36 (2011)
3. Bari, M.F., Chowdhury, S.R., Ahmed, R., Boutaba, R., Mathieu, B.: A survey of naming and routing in information-centric networks. IEEE Commun. Mag. **49**(12), 44–53 (2012)
4. Afanasyev, A., Moiseenko, I., Zhang, L.: ndnsim: NDN simulator for NS-3, Technical Report, University of California, Los Angeles (2012)
5. Compagno, A., Conti, M., Gasti, P., Tsudik, G.: Poseidon: Mitigating interest flooding DDoS attacks in named data networking, arXiv preprint:1303.4823 (2013)
6. Afanasyev, A., Mahadevany, P., Moiseenko, I., Uzuny, E., Zhang, L.: Interest flooding attack and countermeasures in named data networking. In: Proceedings of IFIP Networking, Brooklyn, New York, USA (2013)
7. Fotiou, N., Marias, G., F., Polyzos, G., C.: Fighting spam in publish/subscribe networks using information ranking. In: 6th EURO-NF Conference on Next Generation Internet (NGI), pp. 1–6, Paris (2010)
8. Gasti, P., Tsudik, G., Uzun, E., Zhang, L.: DoS and DDoS in named data networking. In: Proceedings of the 22nd International Conference on Computing Communications and Networks. IEEE (2013)
9. Zargar, S., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun. Surv. Tutorials **15**(4), 2046–2069 (2013)
10. You, Y., Zulkernine, M., Haque, A.: A Distributed defense framework for flooding-based DDoS attacks. In: Proceedings of the International Conference on Availability, Reliability and Security, pp. 245–252. IEEE CS Press, Barcelona, Spain (2008)
11. Keromytis, A., Misra, V., Rubenstein, D.: SOS: an architecture for mitigating DDoS attacks. IEEE J. Sel. Areas Commun. **22**(1), 176–188 (2004)